International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Network Congestion and Decongestion: A Case Study of Cellular Network

Felix Udechukwu[1], Mamilus Ahaneku[2], Onyekachi Chibuzo[3], Michael Ezinwa[4]
[1, 2]Department of Electronic Engineering, University of Nigeria Nsukka, Enugu State, Nigeria
[3,4]Enugu State University of Science and Technology Nigeria
E-mail addresses: [1]felix.udechukwu@unn.edu.ng, [2]mamilus.ahaneku@unn.edu.ng, [3]chibuzoonyekachi@yahoo.com,[4]michaelezinwa@yahoo.com

Abstract— *Operators in emerging cellular markets are already facing congestion on their networks in specific areas and at certain times of the day, despite less-than-optimal utilisation levels overall. Utilisation is expected to slowly increase as device penetration grows and data usage increases. The congestion creates a sub-optimal user experience, which leads to service interruptions. Mobile network operators need to understand consumer behaviour and consider related solutions to address this issue because spectrum availability is limited. This article explains congestion and decongestion in a cellular network and also examines options available to operators, including network-related solutions such as TCPs (Transfer Control Protocol) and Verus.*

Keywords— *Congestion, Decongestion, Transfer control protocol, Cellular network.*

I. Introduction

In the context of networks, congestion refers to a network state where a node or link carries so much data that it may deteriorate network service quality, resulting in queuing delay, frame or data packet loss and the blocking of new connections. Congestion can also be seen as a situation in which an increase in data transmissions results in a proportionately smaller throughput [1]. In a congested network, response time slows with reduced network throughput. Congestion occurs when bandwidth is insufficient and network data traffic exceeds the required capacity as provided by the operator. Congestion is usually expected during peak periods as a result of high demand by customers. It could occur during break hours of the day or disaster period. Analysis of traffic flow shows that demand is higher during disaster periods because people want to be current with the situation.

Data packet loss from congestion is partially countered by aggressive network protocol retransmission, which maintains a network congestion state after reducing the initial data load. This can create two stable states under the same data traffic load - one dealing with the initial load and the other maintaining reduced network throughput.

In other words, decongestion is a process of reducing the amount of load on a network. Transfer control protocol (TCP) offers reliable transport of data and utilizes congestion control mechanism. Hence it is of essence to optimize wireless networks for optimal TCP performance in cellular network.

Network decongestion can also be done by a decongestion controller. Here, it is believed that a protocol that relies upon high speed transmission can achieve a better performance and fairness than TCP [3]. The decongestion controller does basically three functions:

- It selects the caravan size
- It picks an appropriate level and type of coding
- It balances transmission rates across destinations

Caravans could range from 1 packet to thousands of packets. In order to get a right caravan, the controller starts with a fixed caravan and begins the transmission loop. The controller doubles the size of the next caravan when a caravan is successfully delivered. If after a predefined timeout there is lesser data in the socket buffer to fill a caravan, the caravan size is reduced to half its previous size. This aids the controller to discover the rate at which data is been generated form the source. As soon as the caravan size has been identified the decongestion controller selects the type and rate of coding to use for each caravan. The type of coding could vary from simple (duplicate transmission) to the complex (LT coding).

II. Congestion Control

Congestion Control is very important in cellular networks. The purpose of congestion control is to ensure network stability and achieve a reasonably fair distribution of the network resources among the users. A congestion management solution should have two major components which are these:

- Mechanised way of suppressing the impact of congestion
- Detection mechanism that can simply detect and trigger the congestion suppression mechanism

In congestion control, the transfer control protocol (TCP) is a very important protocol that can be used to effectively control congestion. It offers a reliable transport of data and applies congestion control.

There are numerous transfer control protocol variations that can be used for congestion control. Some of them are –

- TCP Vegas
- TCP Tahoe
- TCP Reno
- TCP Nice
- BIC (Binary increase congestion control) TCP
- TCP Cubic
- Compound TCP
- Binomial Congestion Control
- Equation based rate Control
- Sprout

Felix Udechukwu, Mamilus Ahaneku, Onyekachi Chibuzo, and Michael Ezinwa, “Network congestion and decongestion: A case study of cellular network,” *International Research Journal of Advanced Engineering and Science*, Volume 3, Issue 3, pp. 148-151, 2018.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

TCP Vegas - This is a congestion avoidance algorithm that was introduced by Brakmo et al in the year 1994. It works based on the principles of packet delay and not packet loss; it uses round trip time (RTT) values of the connection to detect congestion at an early stage. In TCP Vegas, congestion window does not continue to increase during congestion avoidance rather TCP Vegas detects emergent congestion by comparing the measured throughput to the expected throughput. If the two values are close, the congestion window is increased. In congestion avoidance, its congestion scheme checks every RTT whether network conditions has changed enough to cause an effective change in the congestion window adjustment policy. This TCP compares the expected throughput to the measured actual throughput in other to determine how the size of the congestion window will be measured. Expected throughput is measured as

$$\text{Expected} = \frac{window\,size}{Base\,RTT}$$

Window size is the number of bytes currently in transit.
Actual throughput is calculated as

$$\text{Actual} = \frac{\text{rtt len}}{\text{rtt}}$$

rtt len is the number of bytes transmitted during the last RTT.
rtt is the average RTT of the segments acknowledged during the last RTT

TCP Tahoe –This TCP is based on conservation of packets, once the connection is operating at maximum bandwidth capacity packets are not injected into the network until packet is taken out.

TCP Reno: – This TCP works based on the basic principle of TCP Tahoe. It has a slow start congestion avoidance and the coarse grain re-transmit timer. In Reno, there is an immediate acknowledgement whenever a segment is received. In Reno lost packets are detected very early and that prevents the pipeline from empting itself every time a packet is being lost.

TCP Nice –TCP nice added basically three features to what is obtainable in TCP Vegas. It has more sensitive congestion detection; it can reduce the congestion window below one and finally a multiplicative reduction in response to round trip time.

BIC TCP – Binary increase congestion control TCP is mostly use for high speed network. Its congestion window algorithm is unique; this algorithm uses a binary search to accomplish its task. BIC has a unique window growth function.

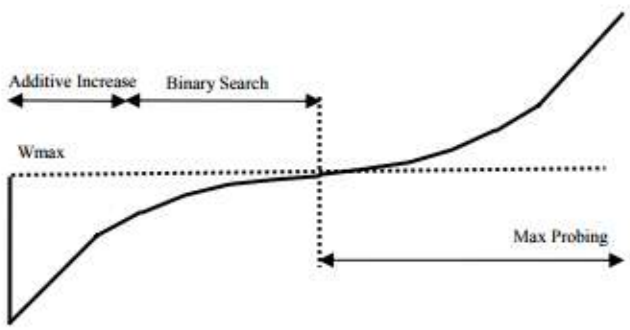

Fig. 1. Window growth function of BIC.

Figure 1 shows the window growth function of BIC. In a packet loss event BIC reduces its window by a multiplicative factor β. BIC performs a binary search using the window size just before the reduction is set to the maximum (W_{max}) and the window size after the reduction is set to the minimum (W_{min}), it jumps to the midpoint between W_{max} and $W_{min.}$

TCP Cubic–Cubic is basically derived from BIC. It made the BIC window control simple and improved RTT-Fairness and TCP-friendliness. The congestion window of Cubic is been determined by this function

$$W_{cubic} = C(t - K)^3 + W_{max}$$

C is the scaling factor
t is known as elapsed time form the previous window reduction
$W_{max}$is the window size before the previous window reduction

$$K = \sqrt[3]{Wmax\ \beta\ C}$$

β represents a constant multiplicative decrease factor applied for window reduction during the time of loss event

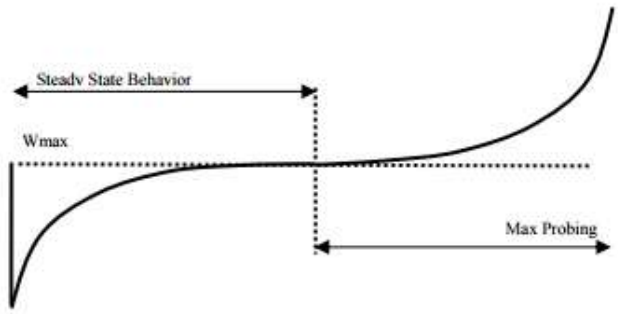

Fig. 2. Window growth function of cubic.

Figure 2 shows the window growth function of Cubic. The window growth function of cubic is a cubic function whose shape is similar to that of BIC. CUBIC is design to enhance and simplify the window control of BIC.

Compound TCP –This is a Microsoft algorithm that was aggressively design to adjust the congestion window for a particular source in other to optimize TCP for large bandwidth connections. It maintains two windows which are Additive increase/multiplicative decrease (AMID) window and a delay based window. The actual sliding window is the total of these two windows, if queuing or delay is noticed the delay window decreases to give room for an increase in the AIMD window. Compound TCP is design to satisfy friendliness requirement and efficiency requirement at the same time.

Binomial Congestion Control – In a binomial congestion control, session begins with a slow start state, here for individual window of packets acknowledged the congestion window size is doubled. Binomial detects congestion by two events which are triple-duplicate ACK and timeout [10].

Equation based rate control –This is a congestion control mechanism that uses a control equation that clearly gives the maximum acceptable sending rate as a function of the recent loss event rate. When there is a response from the receiver the sender adjusts and adapts its sending rate based on the control equation. It should be noted that none of these TCP variants is directly suitable for network conditions where the underlying channels changes at short time scales and here there is no assumption that a link has a fixed capacity [4].

Sprout – Sprout is an end-to-end transport protocol with high throughput and low latency in cellular networks. Sprout observes packet delay by the receiver, using that delay the sender statistically forecasts near-future bandwidth with about 95 percent probability and adjusts its sending rate appropriately.

III. VERUS

This is a congestion control protocol (end to end) that uses delay measurement to quickly react to changes in the capacity of cellular networks without prediction on dynamics of the cellular channel [4]. Cellular channels are not predictable hence verus uses variations in delay to study a delay profile that shows the correlation between the network delay and the amount of data that can be sent without creating network congestion.

Verus works by considering delay feedback from the network as a sign of disagreement and utilizes delay cues to regularly remain in a study mode instead of depending on an assumption which says that delays are self-inflicted [4]. Also, since channel fluctuations happen at unique time-scale; verus uses small ε steps to track and keep record of delay profile.

In [4], Verus was simulated under high contention using OPNET simulator while considering throughput and delay. It was observed that the variation across verus is smaller when compared to TCP cubic and TCP Vegas. This is a sign that verus will quickly adapt and attain high levels of fairness despite mobility.

Studies carried out on verus network in comparison to sprout, TCP vegas and TCP cubic showed the result in figure 1 and figure 2. R is the maximum tolerable ratio between Dmax and Dmin. R could vary to improve the performance of verus. OPNET 14.5 was used for simulation.

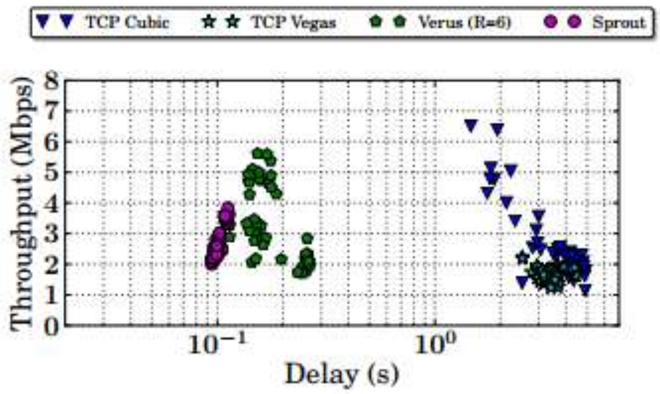

Fig. 3. 3G throughput vs Delay.

Figure 3 shows the relationship between verus, TCP networks and sprout after simulation in a third generation network (3G). From the diagram it’s obvious that verus has a lesser delay time as to compare with that of TCP although sprout network tends to be a bit faster. In terms of throughput verus is much ahead of sprout. Considering verus position in terms of delay and throughput verus is better than the TCPs above and sprout network.

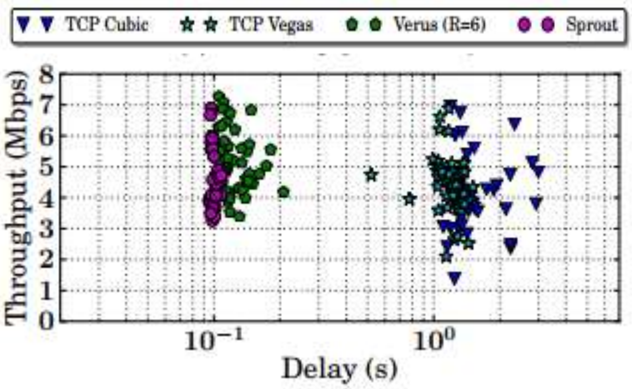

Fig. 4. LTE Throughput vs Delay.

Figure 4 shows the performance of TCP cubic, TCP vegas, Sprout and verus in an LTE network. Verus performed better in Long Term Evolution (LTE) network. From the result in figure 4, verus have a higher throughput than others and also it has a very low delay time which is almost the same with sprout.

Verus also uses small ε steps to track fast changes and delay profile updates to track slower changes. Verus borrows a number of features from legacy TCP variants, such as slow start and multiplicative decrease, but changes the way it maintains the sending window. Legacy TCP uses additive increase and increases the congestion window (CWMD) size by 1/CWND, i.e. increasing the congestion window by one packet when it successfully received a full window. This process can be slow. In contrast, Verus increases/decreases the sending window at each ε ms epoch and adapts to the changing cellular channel by rapidly increasing the sending window when the channel conditions allows for more packets. Similarly, Verus seeks to reduce the sending window even before packet losses whereas TCP can only decrease the congestion window through an aggressive multiplicative decrease after a loss.

Verus and Legacy TCP: Comparative Advantages

Verus borrowed slow start and multiplicative decrease but changes the way it maintains the sending windows. It seeks to reduce the sending window even before packet losses whereas TCP can only decrease the congestion window through an aggressive multiplicative decrease after a loss.

Verus adapts to competing traffic and to swiftly changing network conditions. In cellular network verus achieves higher throughput than TCP cubic and also maintains a lower end to end delay.

The main goal of verus is to avoid congestion by maintaining an appropriate sending window W over a period equal to the estimated network Round Trip Time (RTT). It does this by replacing the additive increase with a series of small E steps to adapt quickly to channel fluctuations.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

IV. Conclusion

This paper talked about numerous Transfer Control protocol and other congestion control networks and mechanism, it compares TCP network to that of Verus and from all indications Verus seems to perform better than most TCP congestion control networks.

References

[1] Sandvine Intelligent Broadband Network, Network Congestion Management: Consideration and Technique, 2015.
[2] Sarah Landstrom, Congestion Control in Network, 2005.
[3] Barath Raghavan and Alex C. Snoeren, "Decongestion control," University of California, San Diego (barath,snoeren)@cs.ucsd.edu.
[4] YasirZaki, Thomas Potsch, Jay Chen, Lakshminarayanan, Carmelita Gorg, *Adaptive Congestion Control for Unpredictable Cellular Networks*, SIGCOMM, London, United Kingdom, 2015. ISBN 978-1-4503-3542-3/15/08.
[5] Alepuz, J. Cabrejas, V. Osa, J. F. Monserrat, J. López, V. Soler, Instituto de Telecomunicaciones y Aplicaciones Multimedia, Universitat Politècnica de València, 8G Building - access D - Camino de Vera s/n - 46022 Valencia (Spain).iralbe@iteam.upv.es.
[6] Paolo Lollini, Andrea Bondavalli, Felicita Di Giandomenico, and S. Porcarelli, "Congestion analysis during outage, congestion treatment and outage recovery for simple GPRS network," *Proceedings. ISCC 2004. Ninth International Symposium on Computers and Communications*, 2004.
[7] Kun Tan, Jingmin Song, Qian Zhang, Murari Sridharan, "A compound TCP approach for high speed and long distance network," *Proceedings IEEE INFOCOM, 25TH IEEE International Conference on Computer Communications INFOCOM*, 2006.
[8] Kun Tan Jingmin Song, Qian Zhang, and Murari Sridharan, "Compound TCP: A scalable and TCP-Friendly congestion control for high-speed networks," in *4th International workshop on Protocols for Fast Long-Distance Networks (PFLDNet)*, 2006.
[9] Deepak Bansal and Hari Balakrishnan, "Binomial congestion control algorithm," *IEEE INFOCOM*, 2001
[10] Y. Richard Yang and Simon S. Lam, "Analysis of binomial congestion control," Department of Computer Science the University of Texas at Austin, 2000.
[11] D. Bansal and H. Balakrishnan, "TCP-Friendly Congestion Control for Real Time Streaming Applications," *MIT Technical Report*, Massachusetts Institute of Technology, USA, 2000.
[12] J. Padhye, V. Firoiu, and D. Towsley, and Jim Kurose, "Modelling TCP throughput: A simple model and empirical validation," in *Proceedings of ACM SIGCOMM*, 1998.
[13] Sally Floyd, Mark Handley, Jitendra Padhye, and Jorg Widmer, "Equation based congestion control for unicast applications," *SIGCOMM*, 2000.
[14] Abdolkarim Mardanian Dehkordi, Vahid Tabataba Vakili, "An equation based rate control for adaptive video streaming over cellular network," *2009 International Conference on Machine Learning and Computing*, IPCSIT vol. 3, 2011.
[15] Arun Venkataramani, Ravi Kokku, and Mike Dahlin, "TCP Nice: A Mechanism for Background Transfer," Laboratory of Advanced System Research, Department of Computer Sciences, University of Texas at Austin, United States of America.
[16] Iniong Rhee and Lisong Xu, "CUBIC: A new TCP-Friendly high speed TCP variant," in *Proceedings of the third PFLDNet Workshop*, France, February 2005.

Felix Udechukwu, Mamilus Ahaneku, Onyekachi Chibuzo, and Michael Ezinwa, "Network congestion and decongestion: A case study of cellular network," *International Research Journal of Advanced Engineering and Science*, Volume 3, Issue 3, pp. 148-151, 2018.